



ELSEVIER

Journal of Computational and Applied Mathematics 128 (2001) 261–279

---

---

JOURNAL OF  
COMPUTATIONAL AND  
APPLIED MATHEMATICS

---

---

www.elsevier.nl/locate/cam

# Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow

David Silvester<sup>a,\*</sup>, Howard Elman<sup>b</sup>, David Kay<sup>c</sup>, Andrew Wathen<sup>d</sup><sup>a</sup>*Department of Mathematics, UMIST, Manchester M60 1QD, UK*<sup>b</sup>*Computer Science Department and Institute of Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA*<sup>c</sup>*School of Mathematical Sciences, University of Sussex, Brighton BN1 9QH, UK*<sup>d</sup>*Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, UK*

Received 14 October 1999; received in revised form 25 January 2000

---

## Abstract

We outline a new class of robust and efficient methods for solving subproblems that arise in the linearization and operator splitting of Navier–Stokes equations. We describe a very general strategy for preconditioning that has two basic building blocks; a multigrid V-cycle for the scalar convection–diffusion operator, and a multigrid V-cycle for a pressure Poisson operator. We present numerical experiments illustrating that a simple implementation of our approach leads to an effective and robust solver strategy in that the convergence rate is independent of the grid, robust with respect to the time-step, and only deteriorates very slowly as the Reynolds number is increased. © 2001 Elsevier Science B.V. All rights reserved.

**Keywords:** Navier–Stokes equations; Incompressible flow; Preconditioning; Multigrid iteration

---

## 1. Introduction

The underlying goal here is to compute solutions of incompressible flow problems modelled by the Navier–Stokes equations in a flow domain  $\Omega \subset \mathbb{R}^d$  ( $d = 2$  or  $3$ ) with a piecewise smooth boundary  $\partial\Omega$ :

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p = 0 \quad \text{in } \mathcal{W} \equiv \Omega \times (0, T), \quad (1.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \mathcal{W}, \quad (1.2)$$

---

\* Corresponding author. Tel.: +44-161-200-3656; fax: +44-161-200-3669.

E-mail address: na.silvester@na-net.ornl.gov (D. Silvester).

together with boundary and initial conditions of the form

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t) \quad \text{on } \tilde{\mathcal{W}} \equiv \partial\Omega \times [0, T]; \quad (1.3)$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega. \quad (1.4)$$

We use standard notation:  $\mathbf{u}$  is the fluid velocity,  $p$  is the pressure,  $\nu > 0$  is a specified viscosity parameter (in a nondimensional setting it is the inverse of the Reynolds number), and  $T > 0$  is some final time. The initial velocity field  $\mathbf{u}_0$  is typically assumed to satisfy the incompressibility constraint, that is,  $\nabla \cdot \mathbf{u}_0 = 0$ . The boundary velocity field satisfies  $\int_{\partial\Omega} \mathbf{g} \cdot \mathbf{n} \, ds = 0$  for all time  $t$ , where  $\mathbf{n}$  is the unit vector normal to  $\partial\Omega$ .

If the aim is to simply compute steady-state solutions of (1.1)–(1.2) then time accuracy is not an issue. In other cases however, having an accurate solution at each time-step is important and the requirements of the time discretisation will be more demanding; specifically, an accurate and unconditionally stable time-discretisation method is necessary to adaptively change the time-step to reflect the dynamics of the underlying flow. We will not attempt to describe the many possibilities — the recent monographs of Gresho and Sani [14] and Turek [29] are worth consulting in this respect — but will restrict attention here to the simplest unconditionally stable approach using a one-stage finite difference discretisation, as given below.

**Algorithm 1.** Given  $\mathbf{u}^0$ ,  $\theta \in [1/2, 1]$ , find  $\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^n$  via

$$\begin{aligned} \frac{(\mathbf{u}^{n+1} - \mathbf{u}^n)}{\Delta t} + \mathbf{u}^* \cdot \nabla \mathbf{u}^{n+\theta} - \nu \nabla^2 \mathbf{u}^{n+\theta} + \nabla p^{n+\theta} &= 0, \\ \nabla \cdot \mathbf{u}^{n+\theta} &= 0 \quad \text{in } \Omega, \\ \mathbf{u}^{n+\theta} &= \mathbf{g}^{n+\theta} \quad \text{on } \partial\Omega. \end{aligned} \quad (1.5)$$

Here  $\mathbf{u}^{n+\theta} = \theta \mathbf{u}^{n+1} + (1 - \theta) \mathbf{u}^n$  and  $p^{n+\theta} = \theta p^{n+1} + (1 - \theta) p^n$ . Note that  $p^0$  is required if  $\theta \neq 1$  so the Algorithm 1 is not self-starting in general. In this case an approximation to  $p^0$  must be computed explicitly by manipulation of the continuum problem, or alternatively it must be approximated by taking one (very small) step of a self-starting algorithm (e.g., with  $\theta = 1$  above).

Algorithm 1 contains the well-known nonlinear schemes of backward Euler and Crank–Nicolson. These methods are given by  $(\mathbf{u}^{n+\theta} = \mathbf{u}^{n+1}, \mathbf{u}^* = \mathbf{u}^{n+1})$ ,  $(\mathbf{u}^{n+\theta} = \mathbf{u}^{n+1/2}, \mathbf{u}^* = \mathbf{u}^{n+1/2})$ , and are first and second-order accurate, respectively. In either case, a nonlinear problem must be solved at every time-level. A well-known linearization strategy is to set  $\mathbf{u}^* = \mathbf{u}^n$  above. This does not affect the stability properties of the time discretisation, but it does reduce the Crank–Nicolson accuracy to first order as  $\Delta t \rightarrow 0$  (the first order accuracy of backward Euler is unchanged). To retain second-order accuracy in a linear scheme the Simo–Armero scheme [24] given by setting  $\mathbf{u}^{n+\theta} = \mathbf{u}^{n+1/2}$  with  $\mathbf{u}^* = (3\mathbf{u}^n - \mathbf{u}^{n-1})/2$  in Algorithm 1 is recommended, see [26] for further details.

Using linearized backward Euler or the Simo–Armero scheme, a frozen-coefficient Navier–Stokes problem (or generalised *Oseen* problem) arises at each discrete time step: given a divergence-free vector field  $\mathbf{w}(\mathbf{x})$  (which we will refer to as the “wind”), the aim is to compute  $\mathbf{u}(\mathbf{x})$  and  $p(\mathbf{x})$  such that

$$\frac{1}{\Delta t} \mathbf{u} + \mathbf{w} \cdot \nabla \mathbf{u} - \nu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \quad (1.6)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (1.7)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega. \quad (1.8)$$

Notice that since (1.6)–(1.8) represents a linear elliptic PDE problem, the existence and uniqueness of a solution  $(\mathbf{u}, p)$  can be established under very general assumptions. The development of efficient methods for solving discrete analogues of (1.6)–(1.8) is the focal point of this work.

An outline is as follows. The spatial discretisation of the generalised Oseen problem is discussed in Section 2. Some standard Krylov iteration methods that are applicable to the (nonsymmetric-) systems that arise after discretisation are briefly reviewed in Section 3. Our general preconditioning approach is then developed in Section 4. This approach builds on our research effort over the last decade on developing effective preconditioners for limiting cases of the Oseen problem (1.6)–(1.8): specifically steady-state Stokes problems ( $\Delta t \rightarrow \infty$ ,  $\mathbf{w} \rightarrow \mathbf{0}$ ) [22]; generalised Stokes problems ( $\mathbf{w} \rightarrow \mathbf{0}$ ), see Silvester and Wathen [23]; and steady Oseen problems ( $\Delta t \rightarrow \infty$ ) [5,6,16]. Some computational experiments that demonstrate the power of our solution methodology are presented in Section 5. Implementation of “pure” multigrid methods seems to be relatively complicated, and performance seems to be (discretisation-) method dependent by comparison. The derivation of analytic bounds on convergence rates for the general preconditioner is an ongoing project which will be treated in a forthcoming paper [7]; in the final section we give a flavour of the analysis by quoting results that we have established in two special cases; potential flow ( $\mathbf{w} = \mathbf{0}$  and  $v = 0$ ) and generalised Stokes flow ( $\mathbf{w} = \mathbf{0}$ ). These cases typically arise using time-stepping methods for (1.1)–(1.2) based on *operator splitting* — showing the inherent generality of the preconditioning approach.

## 2. Spatial discretisation

Given that we would like to solve our model problem (1.6)–(1.8) over irregular geometries, the spatial discretisation will be done using finite element approximation (this also gives us more flexibility in terms of adaptive refinement via a posteriori error control, see, e.g., [17]). We note that the algorithm methodology discussed in the paper applies essentially verbatim to finite difference and finite volume discretisations. In the remainder of this section we briefly review the error analysis associated with mixed finite element approximation of (1.6)–(1.8). For full details see [13].

The weak formulation of (1.6)–(1.8) is defined in terms of the Sobolev spaces  $H_0^1(\Omega)$  (the completion of  $C_0^\infty(\Omega)$  in the norm  $\|\cdot\|_1$ ) and  $L_0^2(\Omega)$  (the set of functions in  $L^2(\Omega)$  with zero mean value on  $\Omega$ ). Defining a velocity space  $\mathbf{X} \equiv (H_0^1(\Omega))^d$  and a pressure space  $M \equiv L_0^2(\Omega)$ , it is easy to see that the solution  $(\mathbf{u}, p)$  of (1.6)–(1.8) satisfies

$$\frac{1}{\Delta t}(\mathbf{u}, \mathbf{v}) + (\mathbf{w} \cdot \nabla \mathbf{u}, \mathbf{v}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{X}, \quad (2.1)$$

$$(\nabla \cdot \mathbf{u}, q) = 0 \quad \forall q \in M, \quad (2.2)$$

where  $(\cdot, \cdot)$  denotes the usual vector or scalar  $L^2(\Omega)$  inner product. Since  $\Omega$  is bounded and connected there exists a constant  $\kappa$  satisfying the continuous *inf-sup* condition:

$$\sup_{\mathbf{w} \in \mathbf{X}} \frac{(p, \nabla \cdot \mathbf{w})}{\|\mathbf{w}\|_1} \geq \kappa \|p\| \quad \forall p \in M. \quad (2.3)$$

Furthermore, since  $\mathbf{w}$  is divergence-free, the bilinear form  $c(\cdot, \cdot)$  given by

$$c(\mathbf{u}, \mathbf{v}) = \frac{1}{\Delta t}(\mathbf{u}, \mathbf{v}) + (\mathbf{w} \cdot \nabla \mathbf{u}, \mathbf{v}) + \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) \quad (2.4)$$

is coercive and bounded over  $\mathbf{X}$ ;

$$c(\mathbf{v}, \mathbf{v}) \geq \nu \|\nabla \mathbf{v}\|^2 \quad \forall \mathbf{v} \in \mathbf{X}, \quad (2.5)$$

$$|c(\mathbf{u}, \mathbf{v})| \leq C_w \|\nabla \mathbf{u}\| \|\nabla \mathbf{v}\| \quad \forall \mathbf{u} \in \mathbf{X}, \forall \mathbf{v} \in \mathbf{X}. \quad (2.6)$$

Existence and uniqueness of a solution to (2.1)–(2.2) then follows from a generalisation of the usual Lax–Milgram lemma [13].

To generate a discrete system we take finite dimensional subspaces  $\mathbf{X}_h \subset \mathbf{X}$  and  $M_h \subset L^2(\Omega)$ , where  $h$  is a representative mesh parameter, and enforce (2.1)–(2.2) over the discrete subspaces (again specifying that functions in  $M_h$  have zero mean to ensure uniqueness). Specifically, we look for a function  $\mathbf{u}_h$  satisfying the boundary condition (1.8), and a function  $p_h \in M_h$  such that

$$\frac{1}{\Delta t}(\mathbf{u}_h, \mathbf{v}) + (\mathbf{w}_h \cdot \nabla \mathbf{u}_h, \mathbf{v}) + \nu(\nabla \mathbf{u}_h, \nabla \mathbf{v}) - (p_h, \nabla \cdot \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{X}_h, \quad (2.7)$$

$$(\nabla \cdot \mathbf{u}_h, q) = 0 \quad \forall q \in M_h, \quad (2.8)$$

where  $\mathbf{w}_h$  represents the interpolant of  $\mathbf{w}$  in  $\mathbf{X}_h$ . Notice that this approximation means that the discrete wind is not actually pointwise divergence-free. From the linear algebra perspective the point is that in the case of the enclosed flow boundary condition (1.3), the discrete convection matrix corresponding to the term  $(\mathbf{w}_h \cdot \nabla \mathbf{u}_h, \mathbf{v})$  is *skew symmetric*.

The well-posedness of (2.7)–(2.8) is not automatic since we do not have an internal approximation. A sufficient condition for the existence and uniqueness of a solution to (2.7)–(2.8) is that the following discrete inf–sup condition is satisfied: there exists a constant  $\gamma$  independent of  $h$  such that

$$\sup_{\mathbf{v} \in \mathbf{X}_h} \frac{(q, \nabla \cdot \mathbf{v})}{\|\nabla \mathbf{v}\|} \geq \gamma \|q\| \quad \forall q \in M_h. \quad (2.9)$$

Note that the semi-norm  $\|\nabla \mathbf{v}\|$  in (2.9) is equivalent to the norm  $\|\mathbf{v}\|_1$  for functions  $\mathbf{v} \in \mathbf{X}$ . The inf–sup condition also guarantees optimal approximation in the sense of the error estimate [13]

$$\|\nabla(\mathbf{u} - \mathbf{u}_h)\| + \|p - p_h\| \leq C \left( \inf_{\mathbf{v} \in \mathbf{X}_h} \|\nabla(\mathbf{u} - \mathbf{v})\| + \inf_{q \in M_h} \|p - q\| \right). \quad (2.10)$$

Note that the constant  $C$  is inversely proportional to the inf–sup constant  $\gamma$  in (2.9).

Since we want to use linear algebra tools it is convenient to express the discrete problem (2.7)–(2.8) as a matrix problem. To do this we introduce discrete operators  $\mathcal{F}: \mathbf{X}_h \mapsto \mathbf{X}_h$  and  $\mathcal{B}: \mathbf{X}_h \mapsto M_h$  defined via

$$(\mathcal{F} \mathbf{v}_h, \mathbf{z}_h) = \frac{1}{\Delta t}(\mathbf{v}_h, \mathbf{z}_h) + (\mathbf{w}_h \cdot \nabla \mathbf{v}_h, \mathbf{z}_h) + \nu(\nabla \mathbf{v}_h, \nabla \mathbf{z}_h) \quad \forall \mathbf{v}_h, \mathbf{z}_h \in \mathbf{X}_h, \quad (2.11)$$

$$(\mathcal{B} \mathbf{v}_h, q_h) = (\mathbf{v}_h, \mathcal{B}^* q_h) = -(\nabla \cdot \mathbf{v}_h, q_h) \quad \forall \mathbf{v}_h \in \mathbf{X}_h, \forall q_h \in M_h, \quad (2.12)$$

so that  $\mathcal{B}^*$  is the adjoint of  $\mathcal{B}$ . With these definitions the discrete problem (2.7)–(2.8) can be rewritten as a matrix system: find  $\mathbf{u}_h$  satisfying the boundary condition (1.8) such that

$$\begin{pmatrix} \mathcal{F} & \mathcal{B}^* \\ \mathcal{B} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_h \\ p_h \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}. \quad (2.13)$$

Furthermore, introducing  $\mathcal{A}: X_h \mapsto X_h$ , satisfying

$$(\mathcal{A}v_h, z_h) = (\nabla v_h, \nabla z_h) \quad \forall v_h, z_h \in X_h, \quad (2.14)$$

the inf-sup inequality (2.9) simplifies to

$$\gamma \|q_h\| \leq \sup_{v_h \in X_h} \frac{(\mathcal{B}v_h, q_h)}{(\mathcal{A}v_h, v_h)^{1/2}} \quad \forall q_h \in M_h. \quad (2.15)$$

It is instructive to express (2.13) and (2.15) in terms of the actual finite element matrices that arise in practice. To this end, let us explicitly introduce the finite element basis sets, say,

$$X_h = \text{span}\{\phi_i\}_{i=1}^n, \quad M_h = \text{span}\{\psi_j\}_{j=1}^m; \quad (2.16)$$

and associate the functions  $u_h, p_h$ , with the vectors  $u \in \mathbb{R}^n, p \in \mathbb{R}^m$  of generalised coefficients,  $p_h = \sum_{j=1}^m p_j \psi_j$ , etc. Defining the  $n \times n$  “convection”, “diffusion” and “mass” matrices  $N_{ij} = (w_h \cdot \nabla \phi_i, \phi_j)$ ,  $A_{ij} = (\nabla \phi_i, \nabla \phi_j)$  and  $G_{ij} = (\phi_i, \phi_j)$ , and also the  $m \times n$  “divergence matrix”  $B_{ij} = -(\nabla \cdot \phi_j, \psi_i)$ , gives the finite element version of (2.13):

$$\begin{pmatrix} \frac{1}{\Delta t} G + N + vA & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \quad (2.17)$$

where the RHS term  $g$  arises from enforcement of the (non-homogeneous) boundary condition on the function  $u_h$ ; see [14, pp. 440–448] for details.

Moreover, introducing the  $m \times m$  pressure “mass” matrix  $Q_{ij} = (\psi_i, \psi_j)$ ; leads to the finite element version of (2.9): for all  $p \in \mathbb{R}^m$ ,

$$\gamma(p^t Q p)^{1/2} \leq \max_u \frac{p^t B u}{(u^t A u)^{1/2}} \quad (2.18)$$

$$= \max_{w=A^{1/2}u} \frac{p^t B A^{-1/2} w}{(w^t w)^{1/2}} \quad (2.19)$$

$$= (p^t B A^{-1} B^t p)^{1/2}, \quad (2.20)$$

since the maximum is attained when  $w = A^{-1/2} B^t p$ . Thus, we have a characterization of the inf-sup constant:

$$\gamma^2 = \min_{p \neq 0} \frac{p^t B A^{-1} B^t p}{p^t Q p}. \quad (2.21)$$

In simple terms it is precisely the square root of the smallest eigenvalue of the preconditioned Schur complement  $Q^{-1} B A^{-1} B^t$ . We also have that

$$(q, \nabla \cdot v) \leq \|q\| \|\nabla \cdot v\| \leq \sqrt{d} \|q\| \|\nabla v\| \quad (2.22)$$

where  $\Omega \subset \mathbb{R}^d$ , and so there also exists a constant  $\Gamma \leq \sqrt{d}$  satisfying

$$\Gamma^2 = \max_{p \neq 0} \frac{p^t B A^{-1} B^t p}{p^t Q p}. \quad (2.23)$$

Note that the tight bound  $\Gamma \leq 1$  was recently established (valid in the case of a conforming approximation space,  $X_h \subset X$ ) by Stoyan [28].

In practice, the inf–sup condition (2.9) is extremely restrictive. Problems arise if the pressure space  $M_h$  is too rich compared to the velocity space  $X_h$ . Although many stable methods have been developed (see [14] for a complete list of possibilities), many natural low order conforming finite element methods like  $Q_1$ – $P_0$  (trilinear/bilinear velocity with constant pressure) are unstable in the sense that pressure vectors  $p \in M_h$  can be constructed for which the inf–sup constant tends to zero under uniform grid refinement. This type of instability can be difficult to detect in practice since the associated discrete systems (2.17) are all nonsingular — so that every discrete problem is uniquely solvable — however they become ill-conditioned as  $h \rightarrow 0$ .

Another issue, which needs to be addressed when applying multigrid solution techniques to convection–diffusion problems of the form

$$c(\mathbf{u}_h, \mathbf{v}) = (\mathbf{f}_h, \mathbf{v}) \quad \forall \mathbf{v} \in X_h, \quad (2.24)$$

(with  $c(\cdot, \cdot)$  given by (2.4)), is that standard approximation methods may produce an unstable, possibly oscillating, solution if the mesh is too coarse in critical regions. In such cases, to give additional stability on coarse meshes used in the multigrid process, the discrete problem (2.24) needs to be stabilised. For example, using a *streamline-diffusion* method, we replace (2.24) by the regularised problem

$$c(\mathbf{u}_h, \mathbf{v}) + \delta(\mathbf{w}_h \cdot \nabla \mathbf{u}_h, \mathbf{w}_h \cdot \nabla \mathbf{v}) = (\mathbf{f}_h, \mathbf{v}) \quad \forall \mathbf{v} \in X_h, \quad (2.25)$$

where  $\delta$  is a locally defined stabilisation parameter, see [15] for further details.

The formulation (2.25) clearly has better stability properties than (2.24) since there is additional coercivity in the local flow direction. The local mesh Péclet number  $P_T^e = \|\mathbf{w}_h\|_{\infty, T} h_T / \nu$  determines the streamline-diffusion coefficient  $\delta_T$  in a given element  $T$  via the “optimal” formula [10];

$$\delta_T = \begin{cases} \frac{1}{2} h_T (1 - \frac{1}{P_T^e}) & \text{if } P_T^e > 1, \\ 0 & \text{if } P_T^e \leq 1, \end{cases} \quad (2.26)$$

where  $h_T$  is a measure of the element length in the direction of the wind.

### 3. Krylov subspace solvers

Let  $\mathcal{L}x = f$  denote a generic linear system of equations. Krylov subspace solution methods start with a guess  $x^{(0)}$  for the solution, with residual  $r^{(0)} = f - \mathcal{L}x^{(0)}$ , and construct a sequence of approximate solutions of the form

$$x^{(k)} = x^{(0)} + p^{(k)} \quad (3.1)$$

where  $p^{(k)}$  is in the  $k$ -dimensional *Krylov space*

$$\mathcal{K}_k(r^{(0)}, \mathcal{L}) = \text{span}\{r^{(0)}, \mathcal{L}r^{(0)}, \dots, \mathcal{L}^{k-1}r^{(0)}\}.$$

In this section, we give a brief overview of properties of Krylov subspace methods for solving the systems arising from the discretizations discussed in the previous section.

Problem (2.17) is nonsymmetric so that algorithms applicable to such problems are of primary concern, but the small Reynolds number limit leads to a symmetric indefinite Stokes problem, and we first briefly discuss this case. It is well-known that for symmetric indefinite problems, the MINRES

algorithm [20] generates iterates of the form (3.1) for which the residual  $r^{(k)}$  has minimal Euclidean norm. It follows that the residuals satisfy

$$\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} \leq \min_{\phi_k(0)=1} \max_{\lambda \in \sigma(\mathcal{L})} |\phi_k(\lambda)|,$$

where the minimum is taken over polynomials  $\phi_k$  of degree  $k$  satisfying  $\phi_k(0)=1$ . This result leads to the following bound on the relative residual norm [18].

**Theorem 3.1.** *If the eigenvalues of  $\mathcal{L}$  are contained in two intervals  $[-a, -b] \cup [c, d]$  with  $a - b = d - c > 0$ , then the residuals generated by MINRES satisfy*

$$\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} \leq 2 \left( \frac{1 - \sqrt{\beta}}{1 + \sqrt{\beta}} \right)^{k/2},$$

where  $\beta = (bc)/(ad)$ .

We apply this result to the Stokes equations in the final section. We also point out that tighter bounds can be established when  $a - b \neq d - c$  and  $b, d$  have some asymptotic behaviour, [32,33]. Each step of the computation entails only a matrix–vector product together with a small number, independent of the iteration count, of vector operations (scalar–vector products and inner products), so that the cost per step of the MINRES iteration is low.

For nonsymmetric problems, there is no Krylov subspace solver that is optimal with respect to some error norm for which the cost per step is independent of the iteration count [8,9]. The *generalized minimal residual algorithm* (GMRES) [21] is the most efficient “optimal” solver, producing the unique iterate of the form (3.1) for which the Euclidean norm of the residual is smallest. Step  $k$  requires one matrix–vector product together with a set of  $k$  vector operations, making its cost, in terms of both operation counts and storage, proportional to  $kN$  where  $N$  is the problem dimension. We summarize the main convergence properties of GMRES below. See [4,21] for proofs.

**Theorem 3.2.** *Let  $x^{(k)}$  denote the iterate generated after  $k$  steps of GMRES, with residual  $r^{(k)} = f - \mathcal{L}x^{(k)}$ .*

- (i) *The residual norms satisfy  $\|r^{(k)}\|_2 = \min_{\phi_k(0)=1} \|\phi_k(\mathcal{L})r^{(0)}\|_2$ .*
- (ii) *If  $\mathcal{L} = X\Lambda X^{-1}$  is diagonalizable, where  $\Lambda$  is the diagonal matrix of eigenvalues of  $\mathcal{L}$ , then*

$$\|r^{(k)}\|_2 \leq \|X\|_2 \|X^{-1}\|_2 \min_{\phi_k(0)=1} \max_{\lambda_j} |\phi_k(\lambda_j)| \|r^{(0)}\|_2.$$

Assertions (i) and (ii) follow from the optimality of GMRES with respect to the residual norm. Assertion (i) guarantees that GMRES will solve any nonsingular problem provided that the dimensions of the Krylov space is large enough. This differentiates GMRES from most other nonsymmetric Krylov subspace methods.

The GMRES iterate is computed as in (3.1) with  $p^{(k)}$  of the form  $p^{(k)} = V_k y^{(k)}$ , where  $V_k$  is a matrix whose columns form an orthogonal basis for  $\mathcal{K}_k$ . The construction of the orthogonal basis is what makes the cost per step high, but once such a basis is available, the iterate with smallest residual norm can be computed cheaply. See [21] for details. Nonoptimal methods compromise on these points, reducing the cost per step by avoiding the construction of an orthogonal

basis, but thereby making the construction of an optimal iterate too expensive. Numerous methods of this type have been proposed, for example, BiCGSTAB [30], BiCGSTAB ( $\ell$ ) [25], CGS [27], QMR [11].

The results of Theorems 3.1–3.2 indicate that if the eigenvalues of  $\mathcal{L}$  are tightly clustered, then convergence will be rapid. In particular, for MINRES, it is desirable for the sizes of the two intervals (one on each side of the origin) to be as small as possible, and well separated from the origin. For GMRES, Theorem 3.2(ii) suggests that convergence will be fast if the eigenvalues can be enclosed in a region in the complex plane that is small. The spectra of the discrete problems of Section 2 are not well-behaved in this sense, and convergence must be enhanced by preconditioning. That is, we use an operator  $\mathcal{P} \approx \mathcal{L}$  and solve an equivalent system such as  $\mathcal{P}^{-1}\mathcal{L}x = \mathcal{P}^{-1}b$ , with a more favorable distribution of eigenvalues, by Krylov subspace iteration.

We conclude this section with a few general observations concerning preconditioning for both symmetric indefinite and nonsymmetric problems. Sections 4 and 6 discuss and analyze some specific strategies suitable for (2.17). First, we note that preconditioning increases the cost per step, since the matrix–vector product now requires a preconditioning operation, i.e., application of the action of  $\mathcal{P}^{-1}$  to a vector. Thus, for the preconditioner to be effective, the improved convergence speed must be enough to compensate for the extra cost.

The MINRES algorithm can be combined with preconditioning by a symmetric positive-definite operator  $\mathcal{P}$ . Formally, MINRES is then applied to the symmetric matrix  $\hat{\mathcal{L}} = \mathcal{P}^{-1}\mathcal{L}\mathcal{P}^{-T}$ , where  $\mathcal{P} = \mathcal{P}\mathcal{P}^T$ . The error bound analogous to that of Theorem 3.1 is

$$\frac{\|r^{(k)}\|_{\mathcal{P}^{-1}}}{\|r^{(0)}\|_{\mathcal{P}^{-1}}} \leq 2 \left( \frac{1 - \sqrt{\beta}}{1 + \sqrt{\beta}} \right)^{k/2}, \quad (3.2)$$

where the intervals defining  $\beta$  now come from the eigenvalues of the preconditioned operator  $\hat{\mathcal{L}}$ . Thus, we seek a preconditioner for which the computation of the action of  $\mathcal{P}^{-1}$  is inexpensive, and for which the eigenvalues of  $\hat{\mathcal{L}}$  are tightly clustered, leading to smaller  $\beta$ . Note also that the norm in (3.2) is now different; for further details see [23]. It is also possible to apply the QMR algorithm to symmetric indefinite problems (with comparable complexity to that of MINRES). In this case a symmetric indefinite preconditioner can be used [12].

For nonsymmetric problems, there is some flexibility in how the preconditioned problem may be formulated, with three possible different “orientations”:

$$\begin{aligned} \text{Left orientation} \quad & [\mathcal{P}^{-1}\mathcal{L}][x] = [\mathcal{P}^{-1}f]; \\ \text{Two-sided orientation} \quad & [\mathcal{P}_1^{-1}\mathcal{L}\mathcal{P}_2^{-1}][\mathcal{P}_2x] = [\mathcal{P}_2^{-1}f]; \\ \text{Right orientation} \quad & [\mathcal{L}\mathcal{P}^{-1}][\mathcal{P}x] = [f]. \end{aligned}$$

The two-sided orientation depends on having an explicit representation of the preconditioner in factored form  $\mathcal{P} = \mathcal{P}_1\mathcal{P}_2$ . In our experience, there is little difference in the effectiveness of these choices. We tend to prefer the “right” variant, especially for use with GMRES, since the norm being minimized (the Euclidian norm of the residual) is then independent of the choice of the preconditioner.



#### 4. Preconditioning strategy

Our starting point is the discrete system  $\mathcal{L}x = f$  associated with (2.17), which we write in the form

$$\begin{pmatrix} F & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \quad (4.1)$$

so that  $F = (1/\Delta t)G + N + \nu A \in \mathbb{R}^{n \times n}$ , with  $B \in \mathbb{R}^{m \times n}$ . Our preconditioning strategy is based on the assumption that a fast solver (typically based on multigrid) is available for the convection–diffusion system  $Fu = f$ . This leads us to consider a block triangular preconditioning

$$\mathcal{P}^{-1} = \begin{pmatrix} F^{-1} & R \\ 0 & -S^{-1} \end{pmatrix}, \quad (4.2)$$

with matrix operators  $R \in \mathbb{R}^{n \times m}$  and  $S \in \mathbb{R}^{m \times m}$  chosen to provide clustering of the eigenvalues  $\sigma(\mathcal{L}\mathcal{P}^{-1})$  of the right preconditioned system

$$\mathcal{L}\mathcal{P}^{-1} = \begin{pmatrix} I_n & FR - B^t S^{-1} \\ BF^{-1} & BR \end{pmatrix}. \quad (4.3)$$

The specific choice of  $R$  and  $S$  in (4.2) satisfying

$$FR - B^t S^{-1} = O, \quad BR = I_m,$$

that is,  $R = F^{-1}B^t S^{-1}$  with  $S = BF^{-1}B^t$ , is the optimal choice, see [19]. For this choice, it follows from (4.3) that  $\sigma(\mathcal{L}\mathcal{P}^{-1}) = \{1\}$ , and preconditioned GMRES converges to the solution of (4.1) in at most two iterations.

Implementation of a right preconditioner for GMRES requires the solution of a system of the form  $\mathcal{P}y = r$  at every step. (QMR also requires the solution of a system with  $\mathcal{P}^t$ .) With the optimal choice of  $R$  and  $S$  we need to compute the vector  $\begin{pmatrix} v \\ q \end{pmatrix}$  satisfying

$$\begin{pmatrix} v \\ q \end{pmatrix} = \begin{pmatrix} F^{-1} & F^{-1}B^t S^{-1} \\ 0 & -S^{-1} \end{pmatrix} \begin{pmatrix} r \\ s \end{pmatrix}, \quad (4.4)$$

for given vectors  $r \in \mathbb{R}^n$ , and  $s \in \mathbb{R}^m$ . Rewriting (4.4) shows that the optimal preconditioner is defined by a two-stage process:

$$\begin{aligned} &\text{Solve for } q: Sq = -s; \\ &\text{Solve for } v: Fv = r - B^t q. \end{aligned} \quad (4.5)$$

To get a practical method, we modify the preconditioning process (4.5) by replacing the matrix operators  $S = BF^{-1}B^t$  and  $F$ , by approximations  $S_*$  and  $F_*$  respectively, designed so that the preconditioned Oseen operator has a tightly clustered spectrum. We are particularly interested in operators  $S_*$  and  $F_*$  derived from multigrid computations such that  $\sigma(SS_*^{-1}) \in \omega_S$  and  $\sigma(FF_*^{-1}) \in \omega_F$  where  $\omega_S$  and  $\omega_F$  represent small convex sets in the right half of the complex plane; ideally, these sets would be independent of the problem parameters  $\nu$ ,  $h$ , and  $\Delta t$ .

The construction of the operator  $F_* \approx F$  is relatively straightforward, see Section 5. The more difficult issue is the construction of a simple multigrid approximation to the Schur complement

$BF^{-1}B^t$ , see, e.g., [29, p. 56]. The approach presented here was developed by Kay and Loghin [16] and represents an improved version of ideas in [6,5].

To motivate the derivation, suppose for the moment that we have an unbounded domain, and that differential operators arising in (1.6)–(1.8) commute:

$$\nabla(1/\Delta t + \mathbf{w} \cdot \nabla - \nu \nabla^2)_p \equiv (1/\Delta t + \mathbf{w} \cdot \nabla - \nu \nabla^2)_u \nabla \quad (4.6)$$

where for any operator  $\Theta$ ,  $\Theta_u$  represents the vector analogue of the scalar operator  $\Theta_p$ . If we further assume that a  $C^0$  pressure approximation is used (so that  $M_h \subset H^1(\Omega)$ ) then we can construct a discrete pressure convection–diffusion operator  $\mathcal{F}_p: M_h \mapsto M_h$  such that

$$(\mathcal{F}_p q_h, r_h) = \frac{1}{\Delta t}(q_h, r_h) + (\mathbf{w}_h \cdot \nabla q_h, r_h) + \nu(\nabla q_h, \nabla r_h) \quad \forall q_h, r_h \in M_h. \quad (4.7)$$

Introducing the  $L_2$ -projection operators  $\mathcal{G}: X_h \mapsto X_h$  and  $\mathcal{Q}: M_h \mapsto M_h$

$$\begin{aligned} (\mathcal{G} \mathbf{v}_h, \mathbf{z}_h) &= (\mathbf{v}_h, \mathbf{z}_h) \quad \forall \mathbf{v}_h, \mathbf{z}_h \in X_h, \\ (\mathcal{Q} q_h, r_h) &= (q_h, r_h) \quad \forall q_h, r_h \in M_h, \end{aligned}$$

then gives the discrete analogue of (4.6)

$$(\mathcal{G}^{-1} \mathcal{B}^*)(\mathcal{Q}^{-1} \mathcal{F}_p) \equiv (\mathcal{G}^{-1} \mathcal{F})(\mathcal{G}^{-1} \mathcal{B}^*). \quad (4.8)$$

A simple rearrangement of (4.8) gives

$$\begin{aligned} (\mathcal{G}^{-1} \mathcal{F})^{-1}(\mathcal{G}^{-1} \mathcal{B}^*) &\equiv (\mathcal{G}^{-1} \mathcal{B}^*)(\mathcal{Q}^{-1} \mathcal{F}_p)^{-1}, \\ \mathcal{F}^{-1} \mathcal{B}^* &\equiv \mathcal{G}^{-1} \mathcal{B}^* \mathcal{F}_p^{-1} \mathcal{Q}. \end{aligned}$$

Hence, assuming that (4.8) is valid, we have an alternative expression for the Schur complement operator  $\mathcal{B} \mathcal{F}^{-1} \mathcal{B}^*: M_h \mapsto M_h$ , namely

$$\mathcal{B} \mathcal{F}^{-1} \mathcal{B}^* \equiv \mathcal{B} \mathcal{G}^{-1} \mathcal{B}^* \mathcal{F}_p^{-1} \mathcal{Q}. \quad (4.9)$$

For equivalence (4.9) to hold, it is necessary for the spaces  $X_h$  and  $M_h$  to be defined with periodic boundary conditions. In the case of an enclosed flow boundary condition like (1.8), the discrete operator  $\mathcal{F}_p$  inherits natural boundary conditions (associated with the space  $M$ ), and in this case (4.9) gives us a starting point for approximating the Schur complement matrix  $S = BF^{-1}B^t$ . Using basis (2.16), we have the approximation

$$BG^{-1}B^tF_p^{-1}Q = P_S \approx S. \quad (4.10)$$

The goal now is to design an efficient implementation of a preconditioner based on (4.10). This requires that fast solvers for the underlying operators  $\mathcal{Q}$  and  $\mathcal{B} \mathcal{G}^{-1} \mathcal{B}^*$  are available: we seek operators  $Q_*$  and  $H_*$  such that there exist constants  $\theta$ ,  $\Theta$ ,  $\lambda$ ,  $\Lambda$  independent of  $h$ , satisfying

$$\theta^2 \leq \frac{p^t Q p}{p^t Q_* p} \leq \Theta^2 \quad \forall p \in \mathbb{R}^m, \quad (4.11)$$

and

$$\lambda^2 \leq \frac{p^t B G^{-1} B^t p}{p^t H_* p} \leq \Lambda^2 \quad \forall p \in \mathbb{R}^m, \quad (4.12)$$

respectively. The practical version of the preconditioner is then defined by replacing the action of  $S^{-1}$  in the first step of (4.5) by the so called  $F_p$  approximation:

$$S_*^{-1} = Q_*^{-1} F_p H_*^{-1}. \quad (4.13)$$

Satisfying (4.11) is straightforward; the simple pressure scaling  $Q_* = \text{diag}(Q)$  does the trick [31]. The upshot is that the action of  $Q^{-1}$  in (4.10) can be approximated very accurately using a fixed (small) number of steps of diagonally scaled conjugate gradient iteration applied to the operator  $Q$ .

Relation (4.12) can also be satisfied using a multigrid approach. The crucial point is that the use of a  $C^0$  pressure approximation space is associated with an alternative inf-sup condition, see, e.g., [1]: for a stable mixed approximation there exists a constant  $\beta$  independent of  $h$ , such that

$$\sup_{v \in X_h} \frac{(v, \nabla q)}{\|v\|} \geq \beta \|\nabla q\| \quad \forall q \in M_h. \quad (4.14)$$

Thus, introducing the pressure Laplacian operator  $\mathcal{A}_p : M_h \mapsto M_h$  such that

$$(\mathcal{A}_p q_h, r_h) = (\nabla q_h, \nabla r_h) \quad \forall q_h, r_h \in M_h,$$

we have that (4.14) is equivalent to

$$\beta(\mathcal{A}_p q_h, q_h)^{1/2} \leq \sup_{v_h \in X_h} \frac{(v_h, \mathcal{B}^* q_h)}{\|v_h\|} \quad \forall q_h \in M_h.$$

Applying the same arguments used to get (2.21) and (2.23), we have a natural characterization in terms of the matrices associated with the finite element basis (2.16):

$$\beta^2 \leq \frac{p^t B G^{-1} B^t p}{p^t A_p p} \leq 1 \quad \forall p \in \mathbb{R}^m. \quad (4.15)$$

In simple terms, for a stable mixed discretisation, the operator  $\mathcal{B}\mathcal{G}^{-1}\mathcal{B}^*$  is spectrally equivalent to the Poisson operator  $\mathcal{A}_p$  defined on the pressure space  $M_h$  (with inherited Neumann boundary conditions) [14, p. 563]. We note in passing that an equivalence of the form (4.15) can also hold in cases when a discontinuous pressure approximation is used (with an appropriately defined matrix operator  $A_p$ ). For example, in the case of well known MAC discretization on a square grid, we have  $A_p = h^{-2} B B^t$  where  $A_p$  is the standard five-point Laplacian defined at cell centres.

The result (4.15) opens up the possibility of using a multigrid preconditioner. In particular, a single multigrid V-cycle with point Jacobi or (symmetric) Gauss–Seidel smoothing defines an approximation  $H_*$ , with spectral bounds

$$\kappa^2 \leq \frac{p^t A_p p}{p^t H_* p} \leq 1 \quad \forall p \in \mathbb{R}^m. \quad (4.16)$$

The combination of (4.15) and (4.16) shows that a simple multigrid cycle can be used as an approximation to  $\mathcal{B}\mathcal{G}^{-1}\mathcal{B}^*$  in the sense that (4.12) holds with constants  $\lambda = \beta\kappa$  and  $\Lambda = 1$ .

To end this section we would like to emphasize the simplicity of the practical implementation of the preconditioner associated with (4.5). The computation of  $q$  in the first stage entails an approximation of the action of  $P_S^{-1}$  defined by (4.10). This is done in three steps; the first is the approximation to the action of the inverse of  $BG^{-1}B^t$  using a multigrid iteration applied to a system with coefficient matrix  $A_p$  (typically representing a Poisson operator with Neumann boundary conditions), the second step is a matrix–vector product involving the discrete convection–diffusion operator  $F_p$ , and

the third step is essentially a scaling step corresponding to the solution of a system with coefficient matrix given by the pressure mass matrix  $Q$ . For the second stage of (4.5), the computation of  $v$  is approximated by a multigrid iteration for the convection–diffusion equation. Clearly, the overall cost of the preconditioner is determined by the cost of a convection–diffusion solve on the velocity space and of a Poisson solve on the pressure space; with multigrid used for each of these, the complexity is proportional to the problem size.

## 5. Computational results

We use  $P_2$ – $P_1$  mixed finite element approximation (see, e.g., [14, p. 462]), that is, we choose spaces

$$\begin{aligned} X_h &= \{v \in H_0^1(\Omega) : v|_T \in \mathbb{P}^2(T) \forall T \in \mathcal{T}_h\}, \\ M_h &= \{q \in H^1(\Omega) : q|_T \in \mathbb{P}^1(T) \forall T \in \mathcal{T}_h\}, \end{aligned}$$

where  $T$  is a triangle in the mesh  $\mathcal{T}_h$ . (This mixed method is shown to be inf–sup stable in [1].) We restrict attention to uniformly refined meshes in this work, analogous results for adaptively refined meshes are given in [16].

We present results for three standard test flow problems below. The time discretization is backward Euler, and the linearization strategy is given by the choice  $\mathbf{u}^* = \mathbf{u}^n$  in Algorithm 1. In all cases we run the time integrator for 15 time-steps, unless the stopping criterion  $\|\mathbf{u}^{n+1} - \mathbf{u}^n\|_2 < 10^{-6}$  is satisfied. We solve the linear system that arises at each discrete time interval using GMRES with the preconditioner  $\mathcal{P}$  that is defined below. The GMRES starting vector for the  $n$ th time-step is always taken to be the previous time-step solution  $(\mathbf{u}^{n-1}, p^{n-1})$ . GMRES iterations are performed until the relative residual is reduced by  $10^{-6}$ .

We will denote the action of a single multigrid V-cycle using a point Gauss–Seidel smoother for the discrete velocity operator  $\mathcal{F}$  in (2.11), by  $F_*^{-1}$ ; where we perform one smoothing sweep before a fine to coarse grid transfer of the residual, and one smoothing sweep after a coarse to fine grid transfer of the correction. For details see e.g., [34]. Similarly we let  $H_*^{-1}$  denote the action of a single multigrid V-cycle using damped Jacobi as a smoother (with damping parameter 0.8) for the pressure Laplacian operator  $\mathcal{A}_p$  in (4.7) (again with a single sweep of pre- and post-smoothing). We comment that although the use of multigrid as a solver for a Laplacian operator is very robust, using a simple multigrid cycle with point smoothing does not generally lead to an efficient solver for the convection–diffusion operator  $\mathcal{F}$  when convection dominates (although the same strategy can still be an effective preconditioner [34]). If we let  $Q_*^{-1}$  denote two diagonally scaled conjugate gradient iterations applied to the discrete pressure identity, then our inverse preconditioner is of the form:

$$\mathcal{P}_*^{-1} = \begin{pmatrix} F_*^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & B^T \\ 0 & -I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & Q_*^{-1} F_p H_*^{-1} \end{pmatrix}.$$

Within the multigrid process we construct prolongation operators using interpolation that is consistent with the order of the velocity/pressure approximation spaces. Furthermore, the restriction operator is the usual transpose of the prolongation, and on the coarsest level ( $h = 1/2$  below) we perform an exact solve. Finally, we emphasize that if the local mesh Péclet number is greater than

Table 1  
 $N_{\Delta t}^h$  for Stokes driven cavity flow

$\Delta t$	$h = 1/4$	$h = 1/8$	$h = 1/16$
0.001	9	10	12
0.1	13	14	14
1	14	15	15
10	14	15	15
1000	14	15	15

unity on any grid, then streamline diffusion is included in the discrete system that is solved (as well as the discrete convection–diffusion problems defining the operator  $F_*^{-1}$ , see (2.26)).

To show the robustness of our solver we report below the maximum number of GMRES iterations required for the tolerance to be satisfied on a given mesh (with a given  $\Delta t$ ) over all time iterations; this maximum iteration count is denoted by  $N_{\Delta t}^h$ .

### 5.1. Stokes: driven cavity flow

We firstly consider the (symmetric-) generalized Stokes problem, associated with a standard driven cavity flow problem defined on a unit domain  $\Omega = (0, 1) \times (0, 1)$ . The associated boundary condition is given by

$$\mathbf{u}(\partial\Omega, t) = \begin{cases} (1, 0) & y = 1, \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

and we “spin-up” to the steady state from the initial condition  $\mathbf{u}(\mathbf{x}, 0) = \mathbf{0}$ .

The performance of our preconditioned method is summarized in Table 1. These iteration counts are consistent with our expectation that the rate of convergence is independent of the degree of mesh refinement, and the size of the time-step. We note that in the limit  $\Delta t \rightarrow \infty$ , the system reduces to a stationary Stokes system in which case we have tight analytic bounds showing the effectiveness of the same preconditioning strategy in a MINRES context, see Section 6.

### 5.2. Navier–Stokes: driven cavity flow

We also consider the Navier–Stokes problem associated with the domain, boundary and initial conditions given above. These results are given in Table 2.

The obvious point to note here is that, as in the Stokes case, the performance is not affected by mesh refinement. (The trend is clearly evident even though the meshes are relatively coarse.) In contrast to the results in the Stokes case it can be seen that as  $\Delta t$  gets larger in Table 2, the iteration counts tend to an asymptotic maximum value. Moreover this maximum value becomes somewhat larger as  $\nu$  is decreased. This behaviour is consistent with our expectations — steady-state iteration counts that are presented in [16] can be seen to slowly increase as the Reynolds number is increased. A complete theoretical explanation is not yet available, but see [7].

Table 2  
 $N_{\Delta t}^h$  for Navier–Stokes driven cavity flow

	$h = 1/4$	$h = 1/8$	$h = 1/16$
$\nu = 1/50$			
$\Delta t = 0.1$	14	15	14
$\Delta t = 1$	14	15	15
$\Delta t = 10$	17	18	18
$\nu = 1/100$			
$\Delta t = 0.1$	14	15	14
$\Delta t = 1$	14	16	16
$\Delta t = 10$	19	21	21
$\nu = 1/200$			
$\Delta t = 0.1$	14	15	14
$\Delta t = 1$	15	18	18
$\Delta t = 10$	23	24	24

### 5.3. Navier–Stokes: backward facing step

We finally consider a Navier–Stokes problem on an L-shaped domain. We start with the coarse (level 0) mesh in Fig. 1, and generate subsequent meshes (i.e., levels 1–3) by successive uniform refinement. The total number of degrees of freedom on the respective levels 1, 2 and 3 are 309, 1092 and 4089, respectively. We again start from a “no flow” initial condition, and impose the following enclosed flow boundary condition:

$$\mathbf{u}(\partial\Omega, t) = \begin{cases} (2y - y^2, 0) & x = -6, \\ (\frac{8}{27}(y+1)(2-y), 0) & x = 16, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

An important point here is that the initial condition is not divergence free in this case; i.e.  $\nabla \cdot \mathbf{u}^0 \neq 0$ . This means that the first time-step is artificial (see [14]) — for small  $\Delta t$  it gives a projection to the discretely divergence-free space so that  $\mathbf{u}^1$  looks like a potential flow field. The fact that we

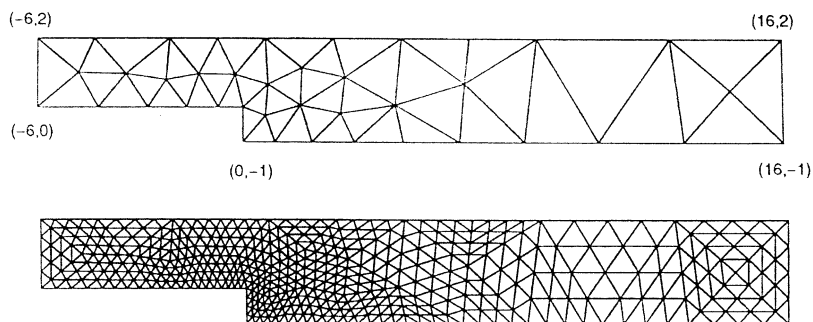
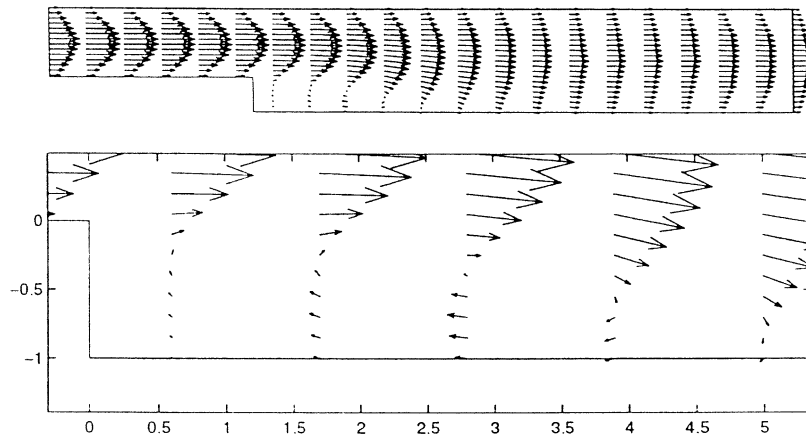


Fig. 1. Coarsest and finest grid triangulations for the backward facing step.

Fig. 2. Velocity solution for  $\nu = 1/200$ .Table 3  
 $N_{\Delta t}^h$  for Navier–Stokes flow over a backward facing step

$\nu = 1/50$	Level 1	Level 2	Level 3
$\Delta t = 0.1$	26	32	33
$\Delta t = 1$	26	32	33
$\Delta t = 10$	26	43	40
$\nu = 1/100$	Level 1	Level 2	Level 3
$\Delta t = 0.1$	26	32	33
$\Delta t = 1$	26	32	33
$\Delta t = 10$	26	51	47
$\nu = 1/200$	Level 1	Level 2	Level 3
$\Delta t = 0.1$	26	32	33
$\Delta t = 1$	26	32	33
$\Delta t = 10$	33	64	59

have a prescribed outflow boundary condition is also emphasized here. Fig. 2 illustrates the computed steady flow (interpolated from the finest mesh) in the case  $\nu = 1/200$ , and shows that the downstream evolution from the inflow to the outflow profile is physically realistic.

The maximum iteration counts are given in Table 3. These results have the same general character as those in Table 2, although the iteration counts for a given  $\nu$  and  $\Delta t$  are increased by a factor of about two. We attribute this difference to the fact that the longer flow domain means that the local mesh Péclet number is relatively large in this case. We remark that for the largest time-step there is a reduction in the iteration count when going from the second to the third level of refinement. Indeed the *average* GMRES iteration counts in the case  $\nu = 1/200$ ,  $\Delta t = 10$  are 27.3, 52.3 and 50.1,

respectively. This phenomenon of increased mesh refinement being correlated with faster convergence is also evident in the steady-state results that are presented in [16].

## 6. Analytic results

For problems where the coefficient matrix is symmetric, specifically whenever  $N = 0$  in (2.17), there is a well-established convergence analysis associated with preconditioners based on the Schur complement approximation (4.10). We outline this theory in this final section.

As discussed in Section 3, MINRES is the optimal Krylov solver in the case of a symmetric coefficient matrix  $\mathcal{L}$ , but it can only be used in conjunction with a symmetric positive definite preconditioning operator  $\mathcal{P}$ . For this reason, in place of the block triangular preconditioner (4.2), we introduce the simpler block diagonal variant

$$\mathcal{P}^{-1} = \begin{pmatrix} F_*^{-1} & 0 \\ 0 & S_*^{-1} \end{pmatrix}, \quad (6.1)$$

and insist that the block-diagonal entries  $F_*$  and  $S_*$  are themselves symmetric. The convergence analysis is based on the following result, which is established by Silvester and Wathen [22].

**Theorem 6.1.** *Assume that the blocks  $F_*$  and  $S_*$  in (6.1) satisfy*

$$\lambda_F \leq \frac{u^t F u}{u^t F_* u} \leq A_F \quad \forall u \in X_h, \quad (6.2)$$

$$\lambda_S \leq \frac{p^t B F^{-1} B^t p}{p^t S_* p} \leq 1 \quad \forall p \in M_h, \quad (6.3)$$

*then the eigenvalues of the preconditioned problem,*

$$\begin{pmatrix} F & B^t \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \lambda \begin{pmatrix} F_* & 0 \\ 0 & S_* \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix}, \quad (6.4)$$

*lie in the union of intervals*

$$E \equiv \left[ \frac{1}{2}(\lambda_F - \sqrt{\lambda_F^2 + 4A_F}), \frac{1}{2}(\lambda_F - \sqrt{\lambda_F^2 + 4\lambda_S\lambda_F}) \right] \cup \left[ \lambda_F, \frac{1}{2}(A_F + \sqrt{A_F^2 + 4A_F}) \right]. \quad (6.5)$$

We now consider two special cases; corresponding to potential flow and generalized Stokes flow, respectively.

### 6.1. Potential flow

In the simplest case of potential flow,  $v = 0$  and  $N = 0$  in (2.17) thus in (6.4) we have that  $F = (1/\Delta t)G$ , and the Schur complement matrix is  $S = \Delta t B G^{-1} B^t$ . Since  $F$  is simply a (scaled) velocity mass matrix, the choice of  $F_* \equiv (1/\Delta t)\text{diag}(G)$  ensures that (6.2) holds with  $\lambda_F$  and  $A_F$  independent of  $h$ . For the Schur complement, we consider a preconditioner corresponding to (4.10)



with  $F_p = (1/\Delta t)Q$ , and with  $BG^{-1}B^t$  replaced by the spectrally equivalent operator  $A_p$ , that is we take

$$P_S = BG^{-1}B^tF_p^{-1}Q \simeq A_pF_p^{-1}Q = \Delta t A_p. \quad (6.6)$$

Bound (4.16) suggests that a practical choice for the preconditioner in (6.3) is  $S_* = \Delta t H_*$  corresponding to a (symmetric) multigrid approximation to the inverse of the pressure Poisson operator  $A_p$ . (With this choice of  $S_*$  bounds (4.15) and (4.16) show that (6.3) holds with  $\lambda_S = \beta^2 \kappa^2$ .) Combining Theorems 3.1 and 6.1 then leads to the following result.

**Theorem 6.2.** *In the case of a potential flow problem, MINRES iteration with a velocity scaling together with a simple multigrid preconditioning for the pressure Poisson operator, converges to a fixed tolerance in a number of iterations that is independent of the mesh size  $h$ , and the time step  $\Delta t$ .*

## 6.2. The generalised Stokes equations

We now consider eigenvalue bounds in the case  $N = 0$  in (2.17) so that  $F = (1/\Delta t)G + \nu A$  in (4.1). Since  $F$  is essentially a scaled vector-Laplacian plus an identity operator, it is well-known that multigrid can be used to generate an approximation  $F_*$  satisfying (6.2). For the Schur complement, we consider a preconditioner corresponding to (4.10) with  $F_p = (1/\Delta t)Q + \nu A_p$ , that is we take

$$\begin{aligned} P_S^{-1} &= (BG^{-1}B^t)^{-1}F_pQ^{-1} \\ &\simeq A_p^{-1}F_pQ^{-1} \\ &\equiv (1/\Delta t)A_p^{-1} + \nu Q^{-1}. \end{aligned} \quad (6.7)$$

The optimality of this combination is well established [3]. Using (6.7) we have that the Rayleigh quotient in (6.3) satisfies

$$\frac{p^t B F^{-1} B^t p}{p^t P_S p} = \frac{p^t B ((1/\Delta t)G + \nu A)^{-1} B^t p}{p^t ((1/\Delta t)A_p^{-1} + \nu Q^{-1})^{-1} p}. \quad (6.8)$$

This shows the importance of the inf-sup condition (2.21) in the limiting case of steady flow — for large  $\Delta t$  the quotient (6.8) reduces to the quotient in (2.21), (2.23), and it follows that (6.3) is satisfied with  $\lambda_S = \gamma^2$  in the steady-state limit  $\Delta t \rightarrow \infty$ . Recent work by Bramble and Pasciak [2] has formally established that for finite  $\Delta t$ , quotient (6.8) is bounded both above and below by constants independent of  $h$  and  $\Delta t$ , although careful consideration is required in the separate cases  $\nu \Delta t < h^2$  and  $\nu \Delta t \geq h^2$ .

Our analysis in Section 4 suggests that a practical version of the generalized Stokes preconditioner is given by (6.1) with:

$$S_* = \frac{1}{\Delta t} H_*^{-1} + \nu Q_*^{-1}. \quad (6.9)$$

The point here is that  $P_S$  is spectrally equivalent to  $S_*$  so that (6.3) is satisfied for the choice (6.9), in which case Theorem 6.1 implies that the intervals defining  $E$  in (6.5) are independent of  $h$  and  $\Delta t$ . This fact can be combined with Theorem 3.1 to establish the following convergence result (corroborated by the iteration counts presented in Section 5.1).

**Theorem 6.3.** *In the case of a generalized Stokes problem, preconditioned MINRES iteration with a simple multigrid cycle approximating a Helmholtz operator for each velocity component and a Poisson operator for the pressure, converges to a fixed tolerance in a number of iterations that is independent of the mesh size  $h$ , and the time step  $\Delta t$ .*

## 7. Conclusion

The Navier–Stokes solution algorithm that is outlined in this work is rapidly converging, and the resulting iteration counts are remarkably insensitive to the mesh size and time-step. The algorithm is also robust with respect to variations in the Reynolds number of the underlying flow. An attractive feature of our approach is that it can be implemented using simple building blocks for solving the two subsidiary problems that arise, namely, a pressure Poisson problem and a scalar convection–diffusion problem. If multigrid is used for each of these then the overall complexity is proportional to the number of degrees of freedom.

## Acknowledgements

This is ongoing research and is supported by the EPSRC via grants GR/K91262 (DS), GR/M59044 (AW) and GR/L05617 (DK) and by the US National Science Foundation via grant DMS9972490.

## References

- [1] M. Bercovier, O.A. Pironneau, Error estimates for finite element method solution of the Stokes problem in the primitive variables, *Numer. Math.* 33 (1977) 211–224.
- [2] J. Bramble, J. Pasciak, Iterative techniques for time dependent Stokes problems, *Math. Appl.* 33 (1997) 13–30.
- [3] J. Cahouet, J.P. Chabard, Some fast 3D finite element solvers for the generalised Stokes problem, *Internat. J. Numer. Methods Fluids* 8 (1988) 869–895.
- [4] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.* 20 (1983) 345–357.
- [5] H. Elman, Preconditioning for the steady state Navier–Stokes equations with low viscosity, *SIAM J. Sci. Comput.* 20 (1999) 1299–1316.
- [6] H. Elman, D. Silvester, Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations, *SIAM J. Sci. Comput.* 17 (1996) 33–46.
- [7] H.C. Elman, D.J. Silvester, A.J. Wathen, Performance and analysis of saddle point preconditioners for the discrete steady-state Navier–Stokes equations, Technical Report UMIACS-TR-2000-54 (2000).
- [8] V. Faber, T.A. Manteuffel, Necessary and sufficient conditions for the existence of a conjugate gradient method, *SIAM J. Numer. Anal.* 21 (1984) 352–362.
- [9] V. Faber, T.A. Manteuffel, Orthogonal error methods, *SIAM J. Numer. Anal.* 24 (1987) 170–187.
- [10] B. Fischer, A. Ramage, D. Silvester, A. Wathen, On parameter choice and iterative convergence for stabilised discretisations of advection–diffusion problems, *Comput. Methods Appl. Mech. Eng.* 179 (1999) 185–202.
- [11] R. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* 60 (1991) 315–339.
- [12] R. Freund, N.M. Nachtigal, A new Krylov-subspace method for symmetric indefinite linear systems, in: Ames, W.F. (Ed.), *Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics*, IMACS, 1994, pp. 1253–1256.

- [13] V. Girault, P.A. Raviart, *Finite Element Methods for Navier–Stokes Equations—Theory and Algorithms*, Springer, Berlin, 1986.
- [14] P.M. Gresho, R. Sani, *Incompressible Flow and the Finite Element Method*, Wiley, Chichester, 1998.
- [15] C. Johnson, The streamline diffusion finite element method for compressible and incompressible flow, in: D.F. Griffiths, G.A. Watson (Eds.), *Numerical Analysis 1989*, Longman Scientific, 1989, pp. 155–181.
- [16] D. Kay, D. Lohin, A Green’s function preconditioner for the steady state Navier–Stokes equations, Oxford University Computing Laboratory Report 99/06, 1999, *SIAM J. Sci. Comput.*, submitted for publication.
- [17] D. Kay, D. Silvester, A posteriori error estimation for stabilised mixed approximations of the Stokes equations, *SIAM J. Sci. Comput.* 21 (2000) 1321–1336.
- [18] V.I. Lebedev, Iterative methods for solving operator equations with a spectrum contained in several intervals, *USSR Comput. Math. and Math. Phys.* 9 (1969) 17–24.
- [19] M.F. Murphy, G.H. Golub, A.J. Wathen, A note on preconditioning for indefinite linear systems, *SIAM J. Sci. Comput.* 21 (2000) 1969–1972.
- [20] C.C. Paige, M.A. Saunders, Solution of sparse indefinite systems of linear equations, *SIAM J. Numer. Anal.* 12 (1975) 617–629.
- [21] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 7 (1986) 856–869.
- [22] D. Silvester, A. Wathen, Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners, *SIAM J. Numer. Anal.* 31 (1994) 1352–1367.
- [23] D. Silvester, A. Wathen, Fast & robust solvers for time-discretised incompressible Navier–Stokes equations, in: D.F. Griffiths, G.A. Watson (Eds.), *Numerical Analysis 1995*, Longman Scientific, 1996, pp. 230–240.
- [24] J. Simo, F. Armero, Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier–Stokes and Euler equations, *Comput. Methods Appl. Mech. Eng.* 111 (1994) 111–154.
- [25] G.L.G. Sleijpen, D.R. Fokkema, BiCGSTAB( $\ell$ ) for linear equations involving unsymmetric matrices with complex spectrum, *Electron. Trans. Numer. Anal.* 1 (1993) 11–32.
- [26] A. Smith, D. Silvester, Implicit algorithms and their linearisation for the transient Navier–Stokes equations, *IMA J. Numer. Anal.* 17 (1997) 527–543.
- [27] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 10 (1989) 36–52.
- [28] G. Stoyan, Towards discrete Veltre decompositions and narrow bounds for inf-sup constants, *Comput. Math. Appl.* 38 (1999) 243–261.
- [29] S. Turek, *Efficient Solvers for Incompressible Flow Problems*, Springer, Berlin, 1999.
- [30] H.A. Van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 10 (1992) 631–644.
- [31] A.J. Wathen, Realistic eigenvalue bounds for the Galerkin mass matrix, *IMA J. Numer. Anal.* 7 (1987) 449–457.
- [32] A. Wathen, B. Fischer, D. Silvester, The convergence rate of the minimal residual method for the Stokes problem, *Numer. Math.* 71 (1995) 121–134.
- [33] A. Wathen, B. Fischer, D. Silvester, The convergence of iterative solution methods for symmetric and indefinite linear systems, in: D.F. Griffiths, D.J. Higham, G.A. Watson (Eds.), *Numerical Analysis 1997*, Longman Scientific, 1997, pp. 230–243.
- [34] P. Wesseling, *An Introduction to Multigrid*, Wiley, New York, 1991.